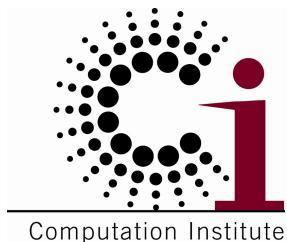


*Strategic Collaborative Initiative:  
Structure of *S. aureus* and metagenome proteins  
using large-scale computational resources*

Argonne – UChicago – Fermilab  
*8<sup>th</sup> Collaboration Meeting*  
Dec 7, 2010

Tobin Sosnick – [trsosnic@uchicago.edu](mailto:trsosnic@uchicago.edu)  
Institute for Biophysical Dynamics, University of Chicago  
<http://sosnick.uchicago.edu/research.html>

Michael Wilde – [wilde@mcs.anl.gov](mailto:wilde@mcs.anl.gov)  
Computation Institute, University of Chicago  
and Argonne National Laboratory  
[www.ci.uchicago.edu/swift](http://www.ci.uchicago.edu/swift)



University of Chicago  
Institute for Biophysical Dynamics

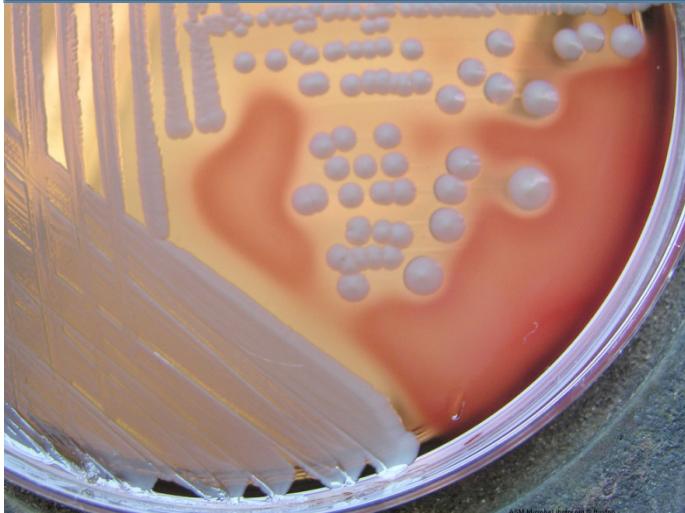


# The Collaboration



- Tobin Sosnick and Michael Wilde, SCI PIs
- Karl Freed and Jinbo Xu, protein folding PIs
- Aashish Adhikari: InsEnds loop algorithm design, implementation, execution and analysis
- ItFix & Raptor: Joe DeBartollo, Glen Hocky, Jian Peng
- Swift team: Justin Wozniak, Mihael Hategan, Luiz Gadhela, Sarah Kenny, David Kelly, Jon Monette, Ben Clifford, Yong Zhao, Allan Espinosa, Zhao Zhang, Ian Foster
- Portal team: Tom Uram, Wenjun Wu, Mark Hereld, Mike Papka
- MRSA Research Center: Robert Daum and Susan Boyle-Vavra

# MRSA: Methicillin-resistant *Staphylococcus aureus*



- Skin infections and numerous complications
- Prevalent in hospitals, prisons
  - Over 18,000 deaths annually
  - \$5 billion healthcare costs
- Multiple strains



ASM MicrobeLibrary.org © Hedetniemi and Liao

# Primary target community

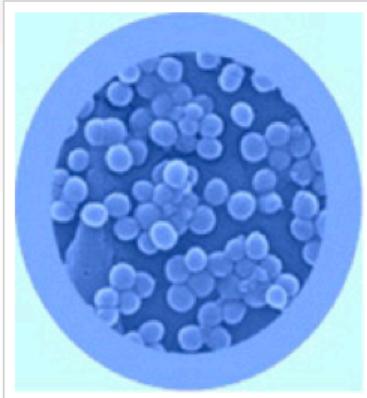


## MRSA Research Center

[Clinical Researchers](#)[Laboratory Researchers](#)[Infection Control Professionals](#)[Patients & Families](#)

[A Message from the PI](#) • [Our Projects](#) • [The Team](#) • [Training Opportunities](#) • [Media Room](#) • [Make a Gift](#) • [News](#) • [Contact Us](#)

### Our Projects



### Core Research Questions

At the MRSA Research Center, we are investigating the following research questions:

- How does MRSA spread? By touching another person? By sharing personal items? How *easily* does MRSA spread?
- Can you get infected from MRSA by touching a surface or object? If so, how long can MRSA last on a surface or object?
- How do you best treat a MRSA infection?
- Why do some people get really sick from MRSA and others don't?
- Why is MRSA so resistant to antibiotics?
- Can we prevent MRSA infection with a vaccine?
- Why is *community-associated* MRSA spreading so quickly?

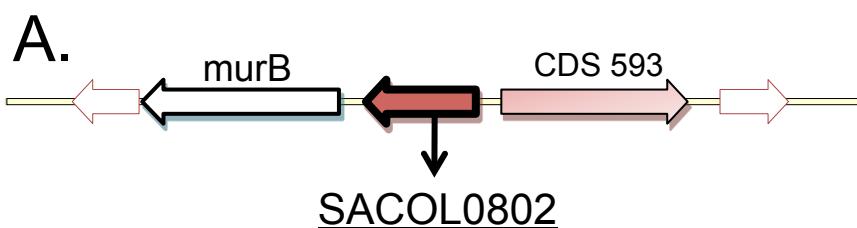


### Studies & Research Projects

#### Modeling MRSA in the Community

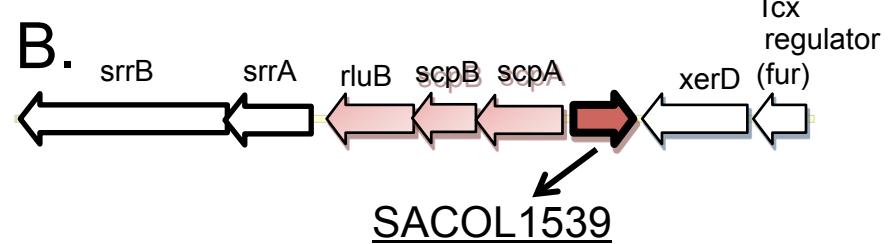
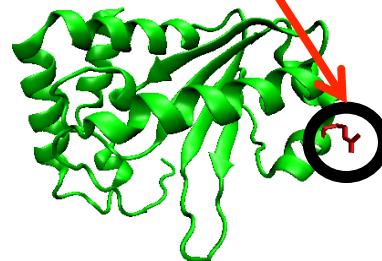
# Hypothetical Mc resistance genes in mecA-negative MRSA

What are they doing?



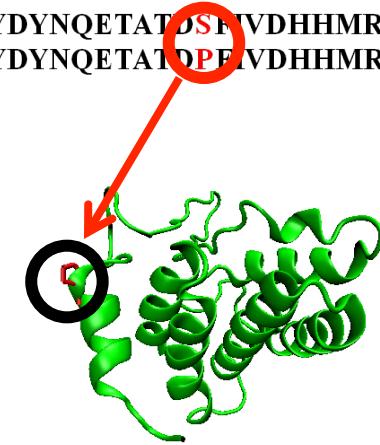
Template : 2nrk (unknown function)  
Confidence: High  
Polymorphism:

Ref ..LHDITSLDE**KR**ENYVGFYRL..  
Var ..LHDITSLDE**KH**ENYVGFYRL..



Template : 2ijq (unknown function)  
Confidence: High  
Polymorphism:

Ref. YDYNQETAT**D**FVDHHMRR..  
Var. YDYNQETAT**P**EIVDHHMRR..



- Current *S. Aureus* campaign:
  - 20 proteins/68 loops x 2,000 jobs per loop, 4.5 hours each = 612,000 hours

# Protein Structure Prediction at UChicago

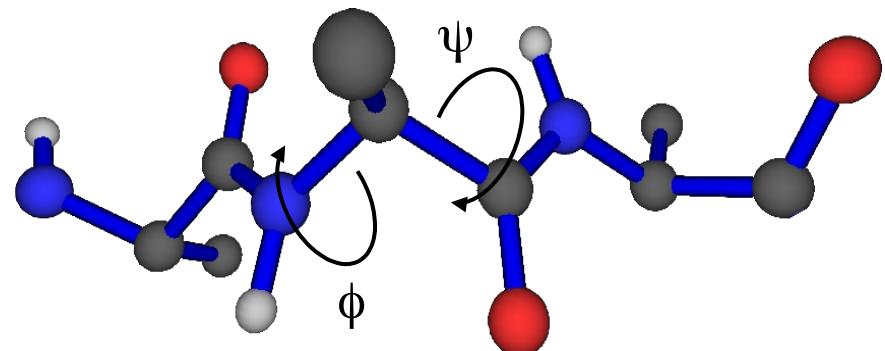


- Joint effort of:
  - Institute for Biophysical Dynamics
    - Tobin Sosnick, Director and PI
    - Aashish Adhikari, Joe DeBartollo, many other PhD students and postdocs
  - Department of Chemistry
    - Karl Freed, PI
  - Toyota Technological Institute
    - Jinbo Xu, PI
  - Computation Institute
    - T. Sosnick, K. Freed, and M. Wilde
- Multi-stage pipeline is used for prediction
  - RAPTOR (Xu) used for “template-based modeling”
  - “OOPS” Open Protein Simulator – “protlib2” used for template-free modeling
  - “Loop modeling” (by Aashish Adhikari) used for detailed simulation of specific loop-like regions that RAPTOR cannot model well. This takes >> CPU time.
  - Applied in CASP9 competition (just completed) and S. Aureus (in progress)
- Swift parallel scripting system enables parallel computing on diverse platforms

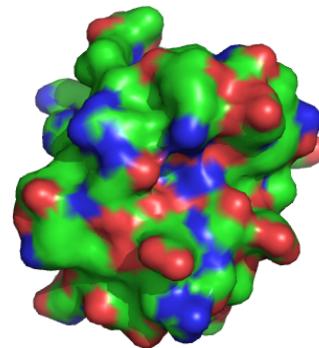
# Themes of protein folding



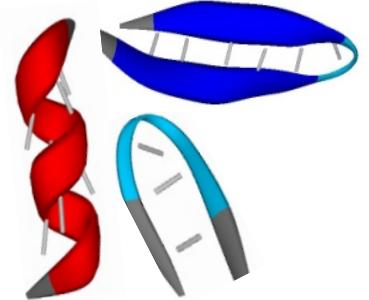
backbone  
twist



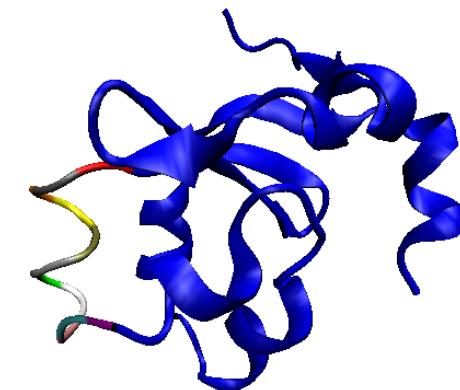
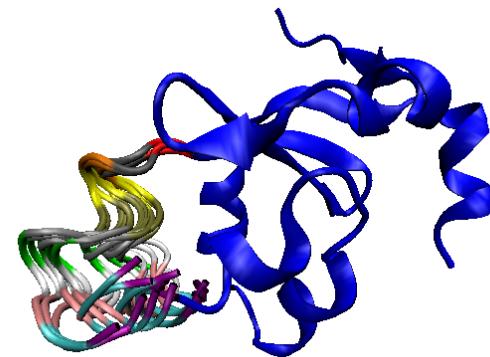
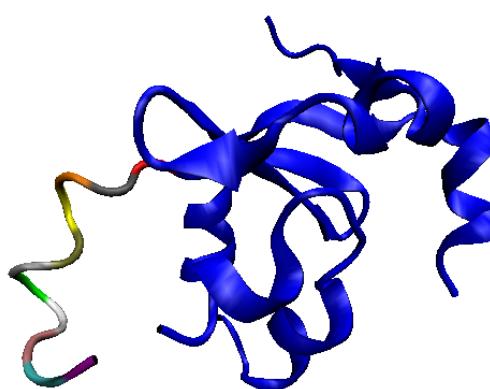
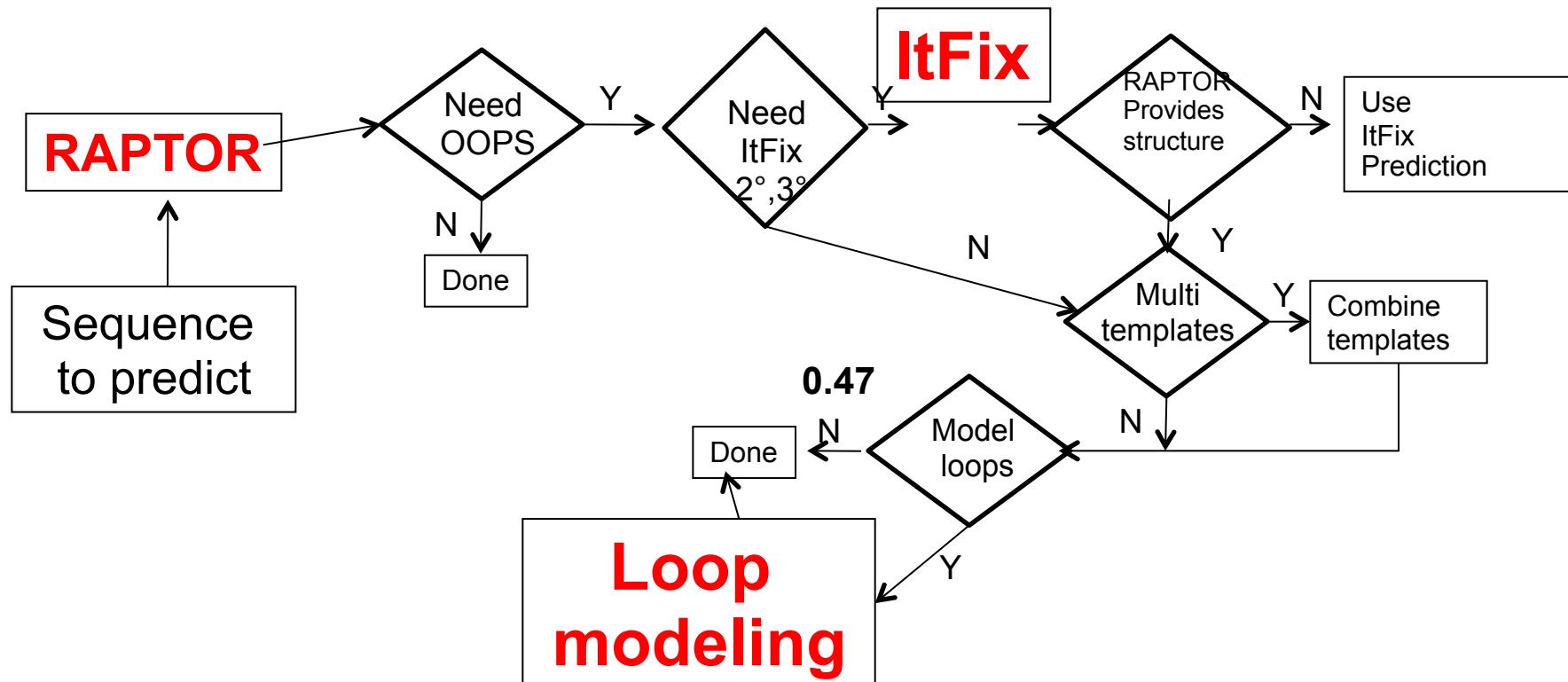
3° structure, packing



2° structure



# Structure prediction pipeline



on TG & OSG

# Loop modeling: “InsEnds”



## Target Sequence

LDGRRRLSVSMDIAAPVSTLKSFVQDETGLPCSKQKLSY EGLFLKDACTLAYYNMNT

## Templates

MSA

LEGQLITLALPLTDQVSVIKA LNGQALTMT---TDQISVIKA LNGQILNITL---ETVSVIKTK LNG--LSLT LNGQLISM-----SVA	-	IKDSNSLAYYNFGP IKDSNTLAFYNVSR FKDSNTLAYYNIGP FKDNNTIAHYNLLN LKDSNSLAYYNILP
-----------------------------------------------------------------------------------------------------------	---	----------------------------------------------------------------------------------------

**INSERTIONS**

NO info in templates

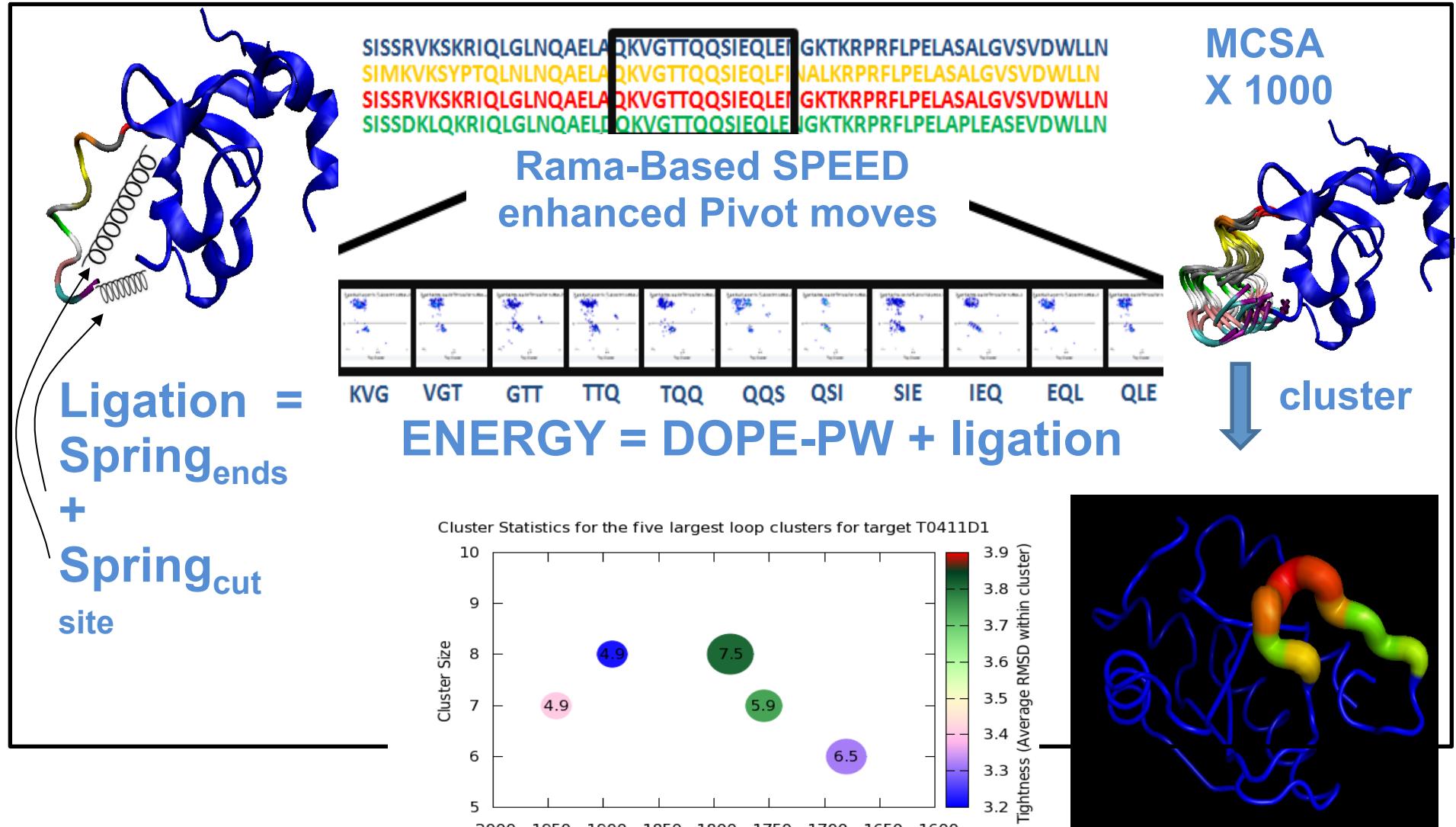
How to model this part?

Template Info

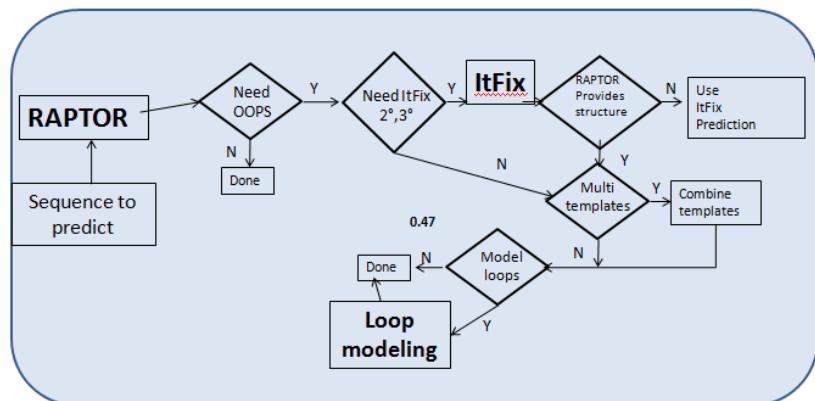
Template Info

Model built from alignment to templates

# InsEnds folding algorithm



# Swift system runs pipeline on diverse parallel systems from central server

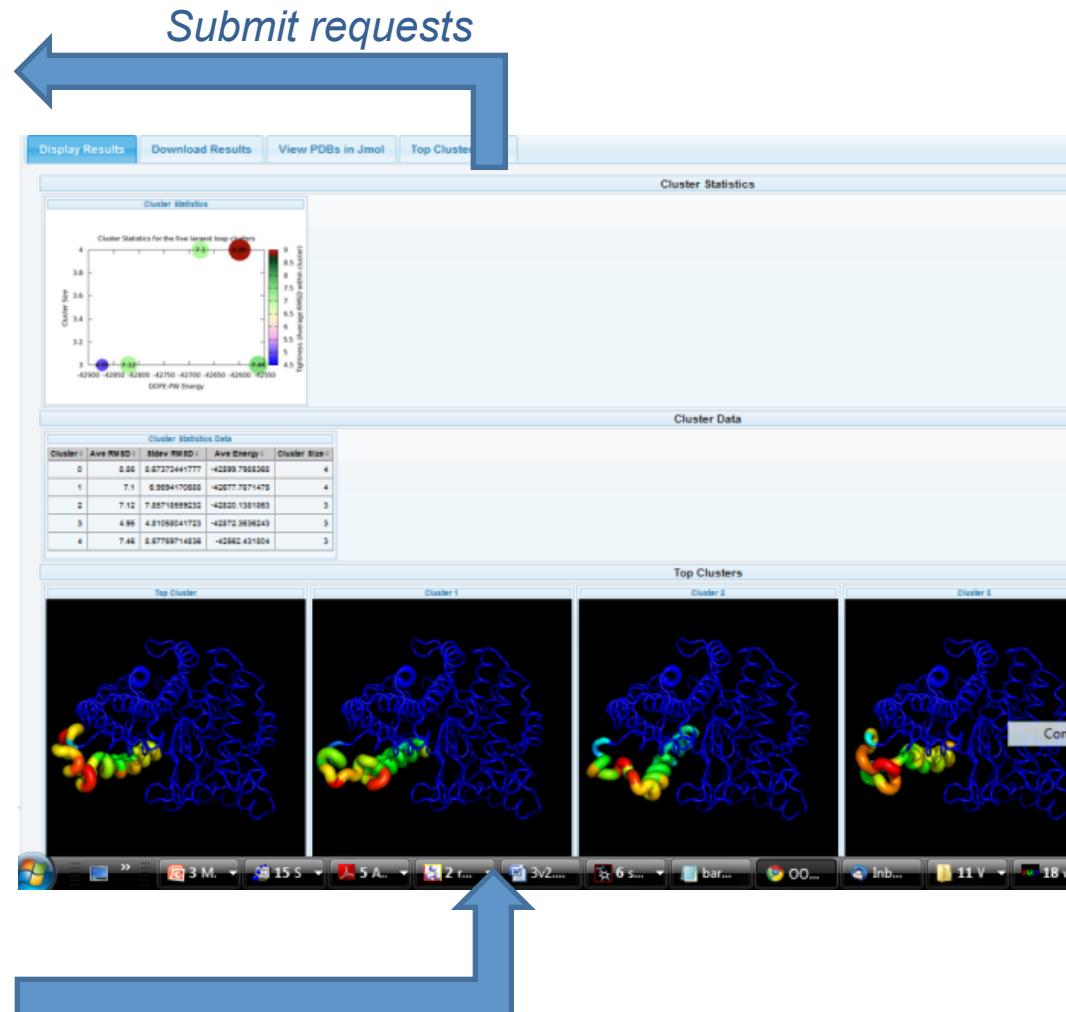


swift,

CI- PADS,Teraport  
TeraGrid - Ranger

IBI and PSD clusters

ALCF BlueGene/P  
Open Science Grid



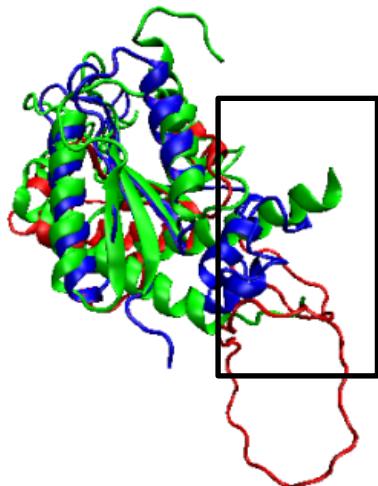
View, download, share results

Portal by MCS Futures Lab  
provides interface for science users

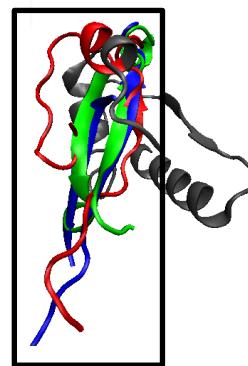
# CASP-9 Results for Loop models



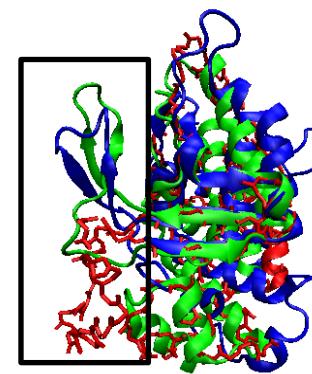
T0585, 45 res. 15.5 Å to 9.1 Å



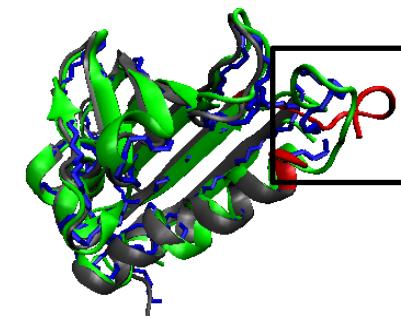
TR606, 30 res.  
(multiple regions simultaneously)  
4.9 Å to 3.8 Å



T0623, 25 res.  
8.2 Å to 6.3 Å  
(excluding tail region)



T0594, 14 res.  
2.2 Å to 1.7 Å



Native  
RaptorX  
Ins&Ends

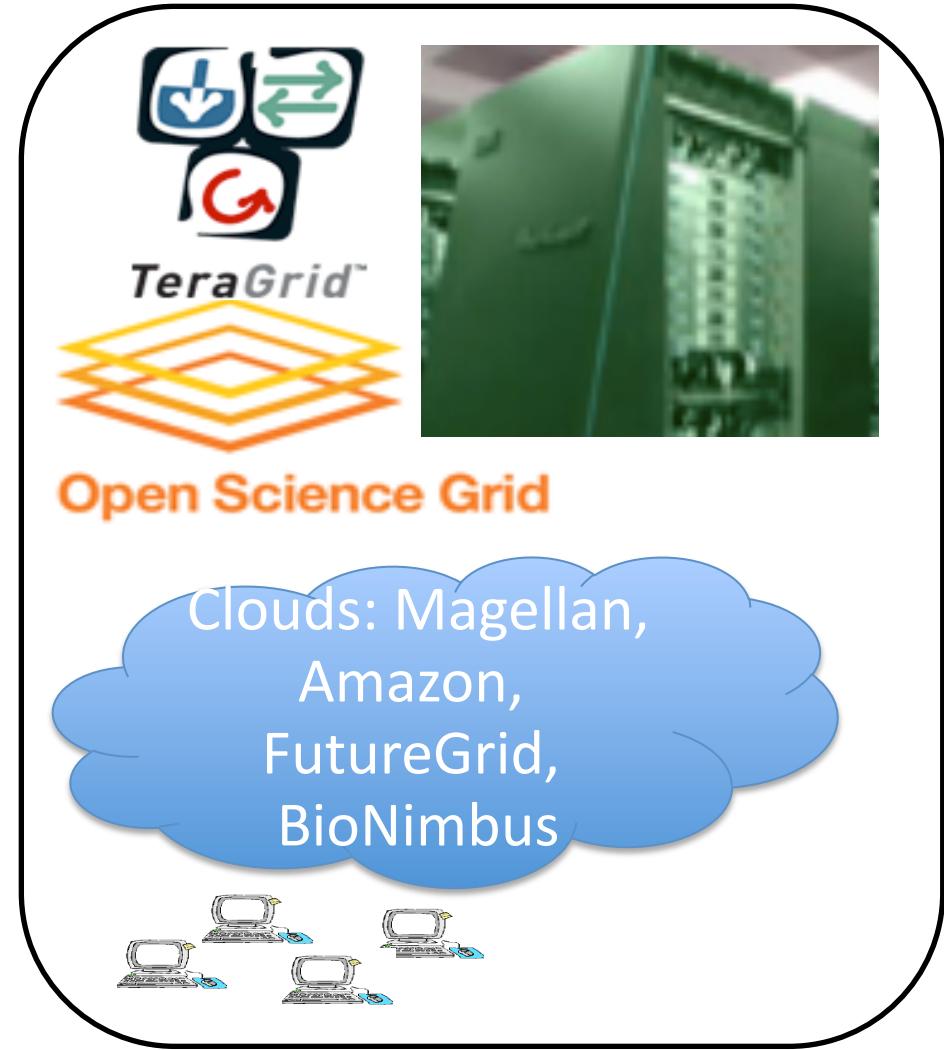
## CASP 9 campaign

150 proteins x 800 jobs/seq x 2.75hours = ~ 330,000 CPU hours

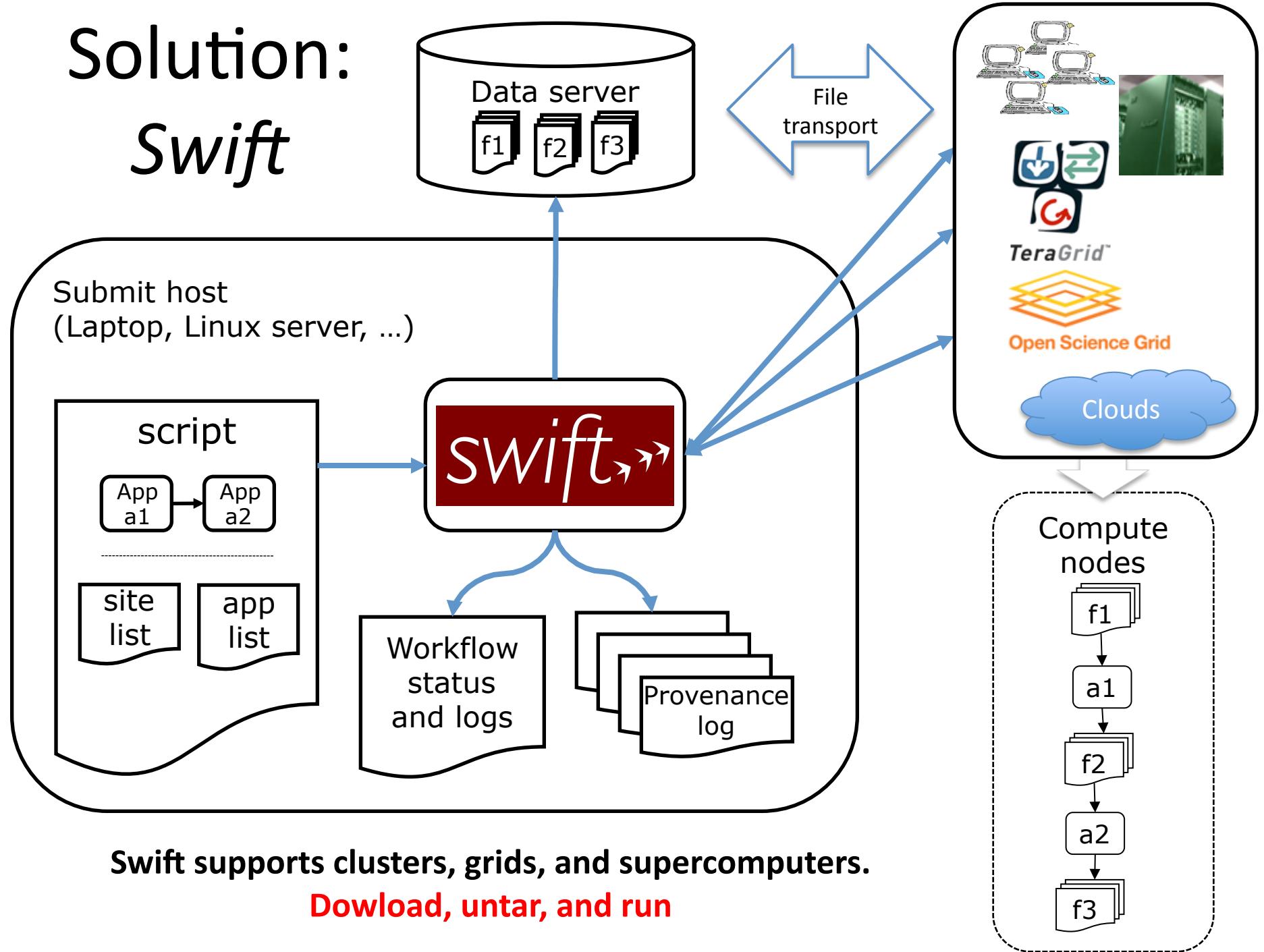
# Challenge: Complexity of parallel computing

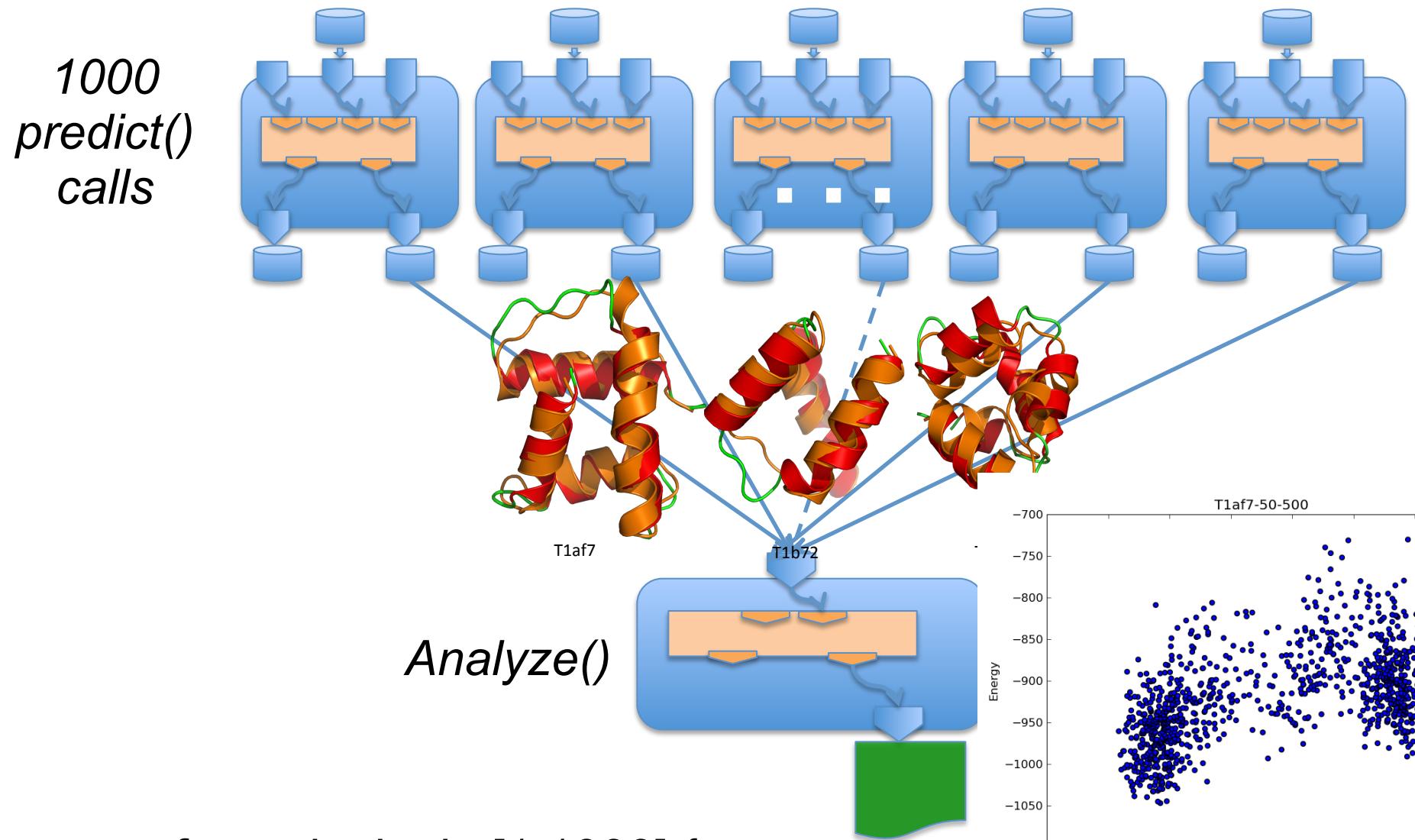


- Parallel distributed computing is HARD
- Swift harnesses diverse resources with simple scripts
- Many applications are well suited to this approach
- Motivates collaboration through libraries of pipelines and sharing of data and provenance



# Solution: *Swift*





```

foreach sim in [1:1000] {
  (structure[sim], log[sim]) = predict(p, 100., 25.);
}
result = analyze(structure)

```

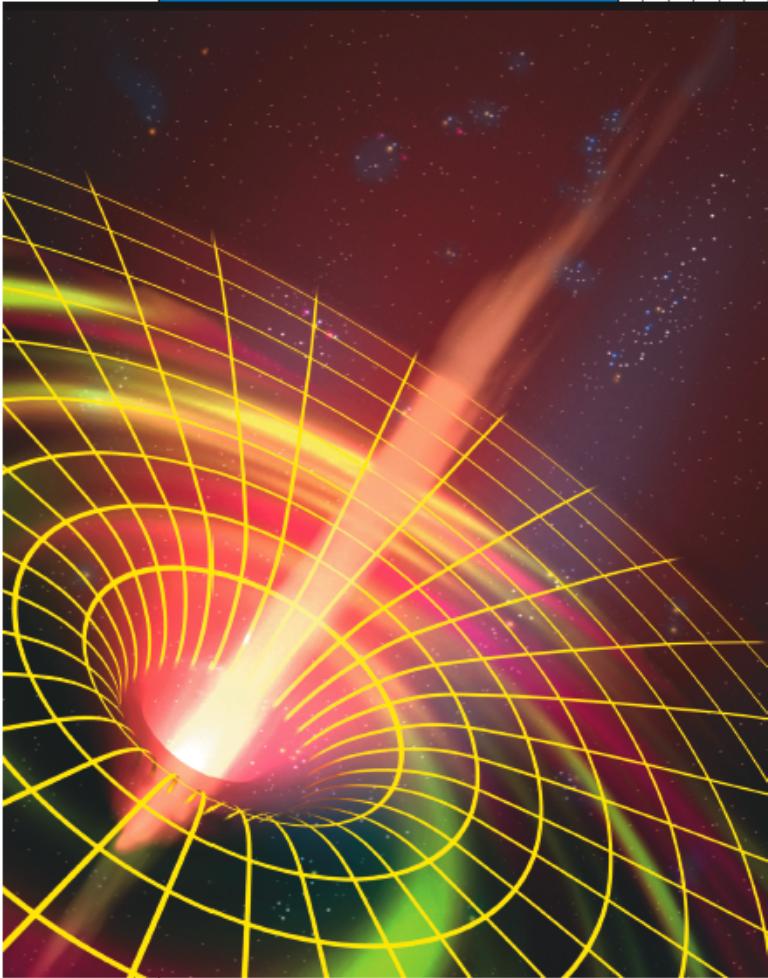
# Powerful parallel prediction loops in Swift



```
1. Sweep( )  
2. {  
3.     int nSim = 1000;  
4.     int maxRounds = 3;  
5.     Protein pSet[ ] <ext; exec="Protein.map">;  
6.     float startTemp[ ] = [ 100.0, 200.0 ];  
7.     float delT[ ] = [ 1.0, 1.5, 2.0, 5.0, 10.0 ];  
8.     foreach p, pn in pSet {  
9.         foreach t in startTemp {  
10.             foreach d in delT {  
11.                 ltFix(p, nSim, maxRounds, t, d);  
12.             }  
13.         }  
14.     }  
15. }  
16. 10 proteins x 1000 simulations x  
17.      3 rounds x 2 temps x 5 deltas  
           = 300K tasks  
18. Sweep();
```



- **Swift is a parallel scripting system for Grids and clusters**
  - for loosely-coupled applications - application and utility programs linked by exchanging files
- **Swift is easy to write:** simple high-level C-like functional language
  - *Small Swift scripts can do large-scale work*
- **Swift is easy to run:** contains all services for running Grid workflow - in one Java application
  - *Untar and run – acts as a self-contained Grid client*
- **Swift is fast:** Karajan provides Swift a powerful, efficient, scalable and flexible execution engine.
  - *Scaling close to 1M tasks – .5M in live science work, and growing*
- **Swift usage is growing:**
  - *applications in neuroscience, proteomics, molecular dynamics, biochemistry, economics, statistics, and more.*



# PARALLEL SCRIPTING FOR APPLICATIONS AT THE PETASCALE AND BEYOND

Michael Wilde, Ian Foster, Kamil Iskra, and Pete Beckman,  
*University of Chicago and Argonne National Laboratory*

Zhao Zhang, Allan Espinosa, Mihael Hategan, and Ben Clifford, *University of Chicago*  
Ioan Raicu, *Northwestern University*

IEEE COMPUTER, Nov 2009

# Science portal for Swift workflow



The image displays two side-by-side screenshots of the OOPS Science Portal interface, showing the progression of a protein simulation workflow.

**Left Screenshot (Input Selection):**

- INPUTS:** A sidebar listing files under "Proteins" (fasta and native) and "rama" (rama, rama\_index, rama\_map, secseq). Examples include T1af7.fasta, T1b72.fasta, T1csp.fasta, etc.
- Run Simulation:** A panel where "Input Proteins" T1ubq and T1csp are selected. It includes fields for "Simulation times" (1), "Starting Temperature" (100), "Time Update Interval" (1), and "Coefficient" (0.1). Buttons for "Upload", "Cancel Uploads", and "Run" are present.
- WORKFLOWS:** A bottom panel with tabs for Console, HTML, CSS, Script, DOM, and Net. The Console tab shows log entries:
  - POST http://communicado.ci.uchicago.edu:8888/SIDGridPortal/Old-JS concat?r.../popup.js (line 379)
  - POST http://communicado.ci.uchicago.edu:8888/SIDGridPortal/Old-JS concat?r.../popup.js (line 379)

**Right Screenshot (Results):**

- INPUTS:** Same file list as the left screenshot.
- Run Simulation:** Same configuration as the left screenshot.
- View Results:** A large panel displaying protein structures. It shows a ribbon model for T1r69 and a green trace model for T1ubq.
- WORKFLOWS:** Same log entries as the left screenshot.

# *ExTENCI* OSG-TeraGrid NSF project supports computing for protein structure prediction



- Maximize resource availability and productivity
  - Run on most OSG sites and many TG sites
  - Optimize choices for MPI vs sequential resources
  - Include local resources (PADS, IBI, PSD, ALCF+)
  - Manageable dataset tracking environment
- Minimize distraction to scientists at increasing scales of execution
  - Fully automated site selection
  - Maximal site (app) availability through dynamic build
- Ability to frequently change & test algorithm (i.e. code)
- Ability to maintain and query provenance
  - Parameters used, code versions used (tie to SVN) – key to scientific experimentation, review, publication, competition
- Provide online prediction services to science community
- Achieving >2,000 core peaks for multi-day workflows for a single user on Open Science Grid

# Acknowledgments



- Swift is supported in part by NSF grants OCI-721939, OCI-0944332, and PHY-636265, NIH DC08638, DOE and UChicago SCI Program
- Structure prediction supported in part by NIH
- The Swift team:
  - Mihael Hategan, Justin Wozniak, Ben Clifford, David Kelly, Allan Espinosa, Ian Foster, Ioan Raicu, Sarah Kenny, Mike Wilde, Zhao Zhang, Yong Zhao
- Science portal:
  - Tom Uram, Wenjun Wu, Mark Hereld, Mike Papka
- Java CoG Kit used by Swift developed by:
  - Mihael Hategan, Gregor Von Laszewski, and many collaborators
- Falkon software
  - developed by Ioan Raicu and Zhao Zhang
- ZeptoOS
  - Kamil Iskra, Kazutomo Yoshii, and Pete Beckman
- Scientific application collaborators and users
  - U. Chicago Open Protein Simulator Group (Karl Freed, Tobin Sosnick, Glen Hocky, Joe Debartolo, Aashish Adhikari)
  - U.Chicago Radiology and Human Neuroscience Lab, (Dr. S. Small)
  - SEE/CIM-EARTH: Joshua Elliott, Meredith Franklin, Todd Muson
  - PTMap: Yingming Zhao, Yue Chen

# To learn more...



- UChicago / TTI protein structure prediction
  - <http://sosnick.uchicago.edu/research.html>
  - <http://home.uchicago.edu/~freed/>
  - <http://ttic.uchicago.edu/~jinbo/>
- Swift system: [www.ci.uchicago.edu/swift](http://www.ci.uchicago.edu/swift)
  - User Guide:
    - <http://www.ci.uchicago.edu/swift/guides/userguide.php>
  - Introductory Swift Tutorials:
    - <http://www.ci.uchicago.edu/swift/docs/index.php>
- UChicago MRSA Research Center
  - [mrsa-research-center.bsd.uchicago.edu/our\\_projects.html](http://mrsa-research-center.bsd.uchicago.edu/our_projects.html)